Assignment 7: Time Series Analysis

Yared S. Asfaw

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on time series analysis.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A07_TimeSeries.Rmd") prior to submission.

The completed exercise is due on Monday, March 14 at 7:00 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

Setting up session

```
# Checking working directory
getwd()
```

```
## [1] "E:/EDA/Environmental_Data_Analytics_2022"
```

```
# Loading packages
library(dplyr)
library(tidyverse)
library(lubridate)
#install.packages("trend")
library(trend)
#install.packages("zoo")
library(zoo)
library(Kendall)
library(tseries)
```

```
# Setting up a ggplot theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right")
# Set theme
theme_set(mytheme)
```

2. Import the ten datasets from the Ozone_TimeSeries folder in the Raw data folder. These contain ozone concentrations at Garinger High School in North Carolina from 2010-2019 (the EPA air database only allows downloads for one year at a time). Import these either individually or in bulk and then combine them into a single dataframe named `GaringerOzone` of 3589 observation and 20 variables.

```
# Importing datasets
GaringerOzone <- dir("data/Raw/Ozone_TimeSeries", full.names = TRUE) %>% map_df(read_csv)
class(GaringerOzone)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

Wrangle

3. Set your date column as a date class.

```
# Formatting the date column to Date class
class(GaringerOzone$Date)
```

```
## [1] "character"
```

```
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format="%m/%d/%Y")
class(GaringerOzone$Date)
```

```
## [1] "Date"
```

4. Wrangle your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE.

```
# Selecting variables of interest
GaringerOzone_select <- select(GaringerOzone,
"Date",Daily_Ozone_Con="Daily Max 8-hour Ozone Concentration",
"DAILY_AQI_VALUE")
head(GaringerOzone_select)
```

```
## # A tibble: 6 x 3
##   Date       Daily_Ozone_Con DAILY_AQI_VALUE
##   <date>               <dbl>           <dbl>
## 1 2010-01-01           0.031              29
## 2 2010-01-02           0.033              31
## 3 2010-01-03           0.035              32
## 4 2010-01-04           0.031              29
## 5 2010-01-05           0.027              25
## 6 2010-01-07           0.033              31
```

5. Notice there are a few days in each year that are missing ozone concentrations. We want to generate a daily dataset, so we will need to fill in any missing days with NA. Create a new data frame that contains a sequence of dates from 2010-01-01 to 2019-12-31 (hint: `as.data.frame(seq())`). Call this new data frame Days. Rename the column name in Days to "Date".

```
# Creating a sequence of dates from 2010-01-01 to 2019-01-01
Days <- as.data.frame(seq(as.Date('2010-01-01'), as.Date('2019-12-31'), by = 1))
head(Days)
```

```
##   seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = 1)
## 1                                                2010-01-01
## 2                                                2010-01-02
## 3                                                2010-01-03
## 4                                                2010-01-04
## 5                                                2010-01-05
## 6                                                2010-01-06
```

```
tail(Days)
```

```
##      seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = 1)
## 3647                                                2019-12-26
## 3648                                                2019-12-27
## 3649                                                2019-12-28
## 3650                                                2019-12-29
## 3651                                                2019-12-30
## 3652                                                2019-12-31
```

```
# Renaming the column name
colnames(Days) <- c("Date")
colnames(Days)
```

```
## [1] "Date"
```

6. Use a `left_join` to combine the data frames. Specify the correct order of data frames within this function so that the final dimensions are 3652 rows and 3 columns. Call your combined data frame GaringerOzone.

```
# Combining the data frames using left_join
GaringerOzone <- left_join(Days, GaringerOzone_select, by= "Date")
head(GaringerOzone)
```

```
##         Date Daily_Ozone_Con DAILY_AQI_VALUE
## 1 2010-01-01           0.031              29
## 2 2010-01-02           0.033              31
## 3 2010-01-03           0.035              32
## 4 2010-01-04           0.031              29
## 5 2010-01-05           0.027              25
## 6 2010-01-06              NA              NA
```

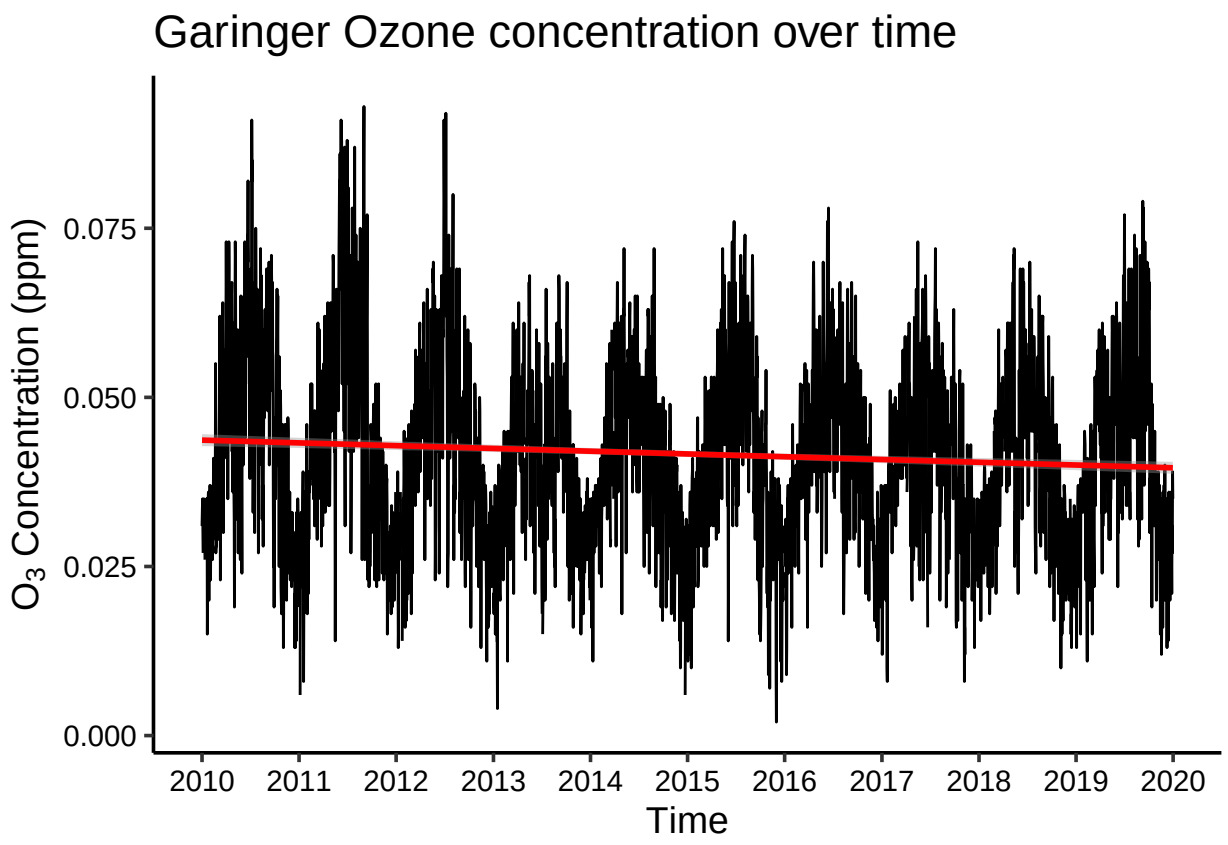
```
tail(GaringerOzone)
```

```
##            Date Daily_Ozone_Con DAILY_AQI_VALUE
## 3647 2019-12-26           0.026              24
## 3648 2019-12-27           0.021              19
## 3649 2019-12-28           0.031              29
## 3650 2019-12-29           0.027              25
## 3651 2019-12-30           0.039              36
## 3652 2019-12-31           0.035              32
```

Visualize

7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
# Plotting the actual ozone concentration over time
ggplot(GaringerOzone,aes(x= Date, y= Daily_Ozone_Con)) +
  geom_line()+
  geom_smooth(color = "red", method = "lm")+
  scale_x_date(date_breaks = "1 year", date_labels = "%Y")+
  xlab("Time") +
  ylab(expression("O"[3]*" Concentration (ppm)"))+
  ggtitle("Garinger Ozone concentration over time")
```



Answer: As per the plot, there is a slight decreasing trend in ozone concentration over time.

Time Series Analysis

Study question: Have ozone concentrations changed over the 2010s at this station?

8. Use a linear interpolation to fill in missing daily data for ozone concentration. Why didn't we use a piecewise constant or spline interpolation?

```
# Filling missing values of Ozone concentration using a linear interpolation
summary(GaringerOzone)
```

```
##       Date            Daily_Ozone_Con   DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200   Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200   1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100   Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04163   Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100   3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300   Max.   :169.00
##                       NA's   :63        NA's   :63
```

```
head(GaringerOzone)
```

```
##         Date Daily_Ozone_Con DAILY_AQI_VALUE
## 1 2010-01-01           0.031              29
## 2 2010-01-02           0.033              31
## 3 2010-01-03           0.035              32
## 4 2010-01-04           0.031              29
## 5 2010-01-05           0.027              25
## 6 2010-01-06              NA              NA
```

```
GaringerOzone$Daily_Ozone_Con <- na.approx(GaringerOzone$Daily_Ozone_Con)
head(GaringerOzone)
```

```
##         Date Daily_Ozone_Con DAILY_AQI_VALUE
## 1 2010-01-01           0.031              29
## 2 2010-01-02           0.033              31
## 3 2010-01-03           0.035              32
## 4 2010-01-04           0.031              29
## 5 2010-01-05           0.027              25
## 6 2010-01-06           0.030              NA
```

```
tail(GaringerOzone)
```

```
##            Date Daily_Ozone_Con DAILY_AQI_VALUE
## 3647 2019-12-26           0.026              24
## 3648 2019-12-27           0.021              19
## 3649 2019-12-28           0.031              29
## 3650 2019-12-29           0.027              25
## 3651 2019-12-30           0.039              36
## 3652 2019-12-31           0.035              32
```

Answer: piecewise constant is not used because we have observations before and after the missing observation from which we can make a better estimation of the missing observation by using the two observations, so that we can avoid over or under estimation while using one neignboring observation only (nearest neighbor value). Spline interpolation is not used because it uses a quandratic function to interpolate rather than drawing a straight line, Linear interpolation uses linear functions for each of the intervals whereas spline interpolation uses low-degree polynomials in each of the intervals. Tus, in this case, for simplicity we prefer to use the linear interpolation method.

9. Create a new data frame called `GaringerOzone.monthly` that contains aggregated data: mean ozone concentrations for each month. In your pipe, you will need to first add columns for year and month to form the groupings. In a separate line of code, create a new Date column with each month-year combination being set as the first day of the month (this is for graphing purposes only)

```
# Aggregating mean Ozone concentrations for each month
 GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Month = month(Date)) %>%
  mutate(Year = year(Date)) %>%
  group_by(Month, Year) %>%
  dplyr::summarise(meanOzone = mean(Daily_Ozone_Con))

# Creating a separate Date column with each month-year combination
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date=my(paste0(Month,"-",Year)))
```

10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

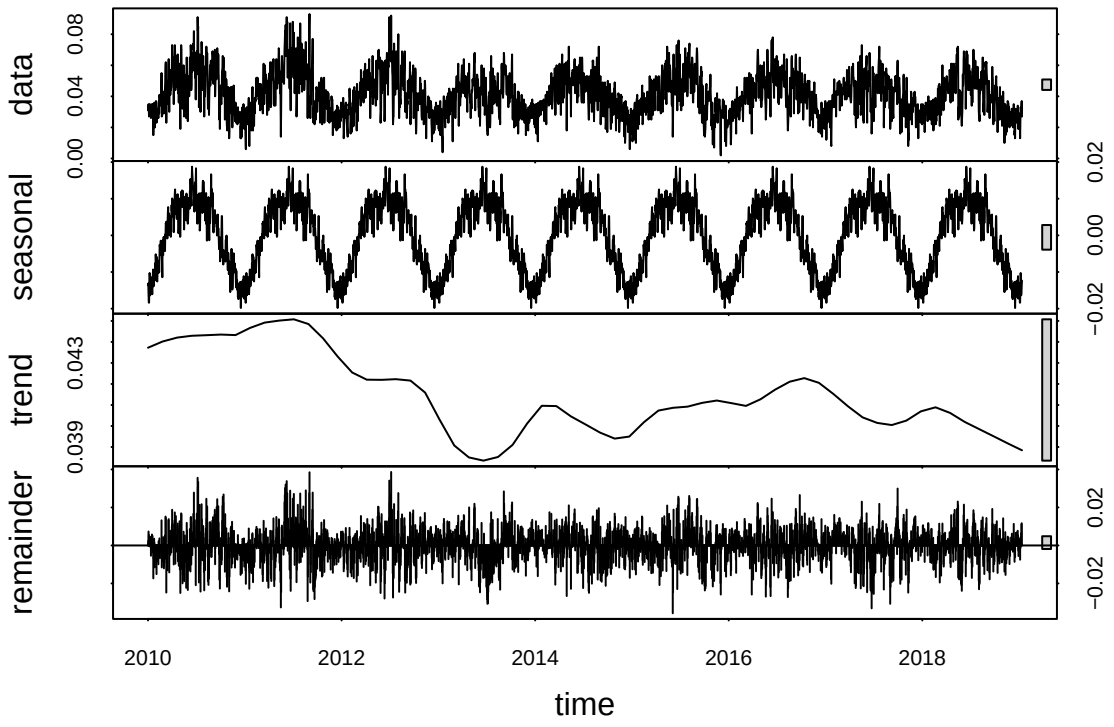
```
# Generating time series object for ozone of daily observations
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily_Ozone_Con, start = c(2010, 01),
end = c(2019, 12), frequency = 365)
```

```
# Generating time series object for ozone of monthly observations
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$meanOzone, start = c(2010, 01),
end = c(2019, 12), frequency = 12)
```

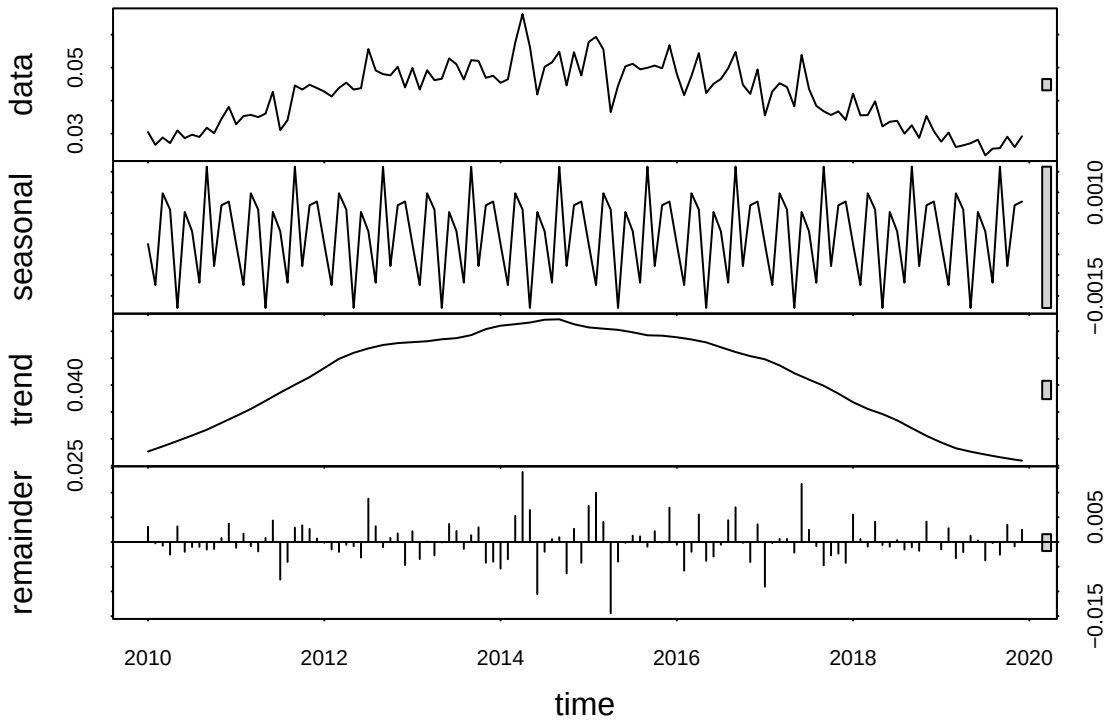
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
# Decomposing and plotting the daily time series object

GaringerOzone.daily.ts_dec <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily.ts_dec)
```




```
# Decomposing and plotting the monthly time series object
GaringerOzone.monthly.ts_dec <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(GaringerOzone.monthly.ts_dec)
```



12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

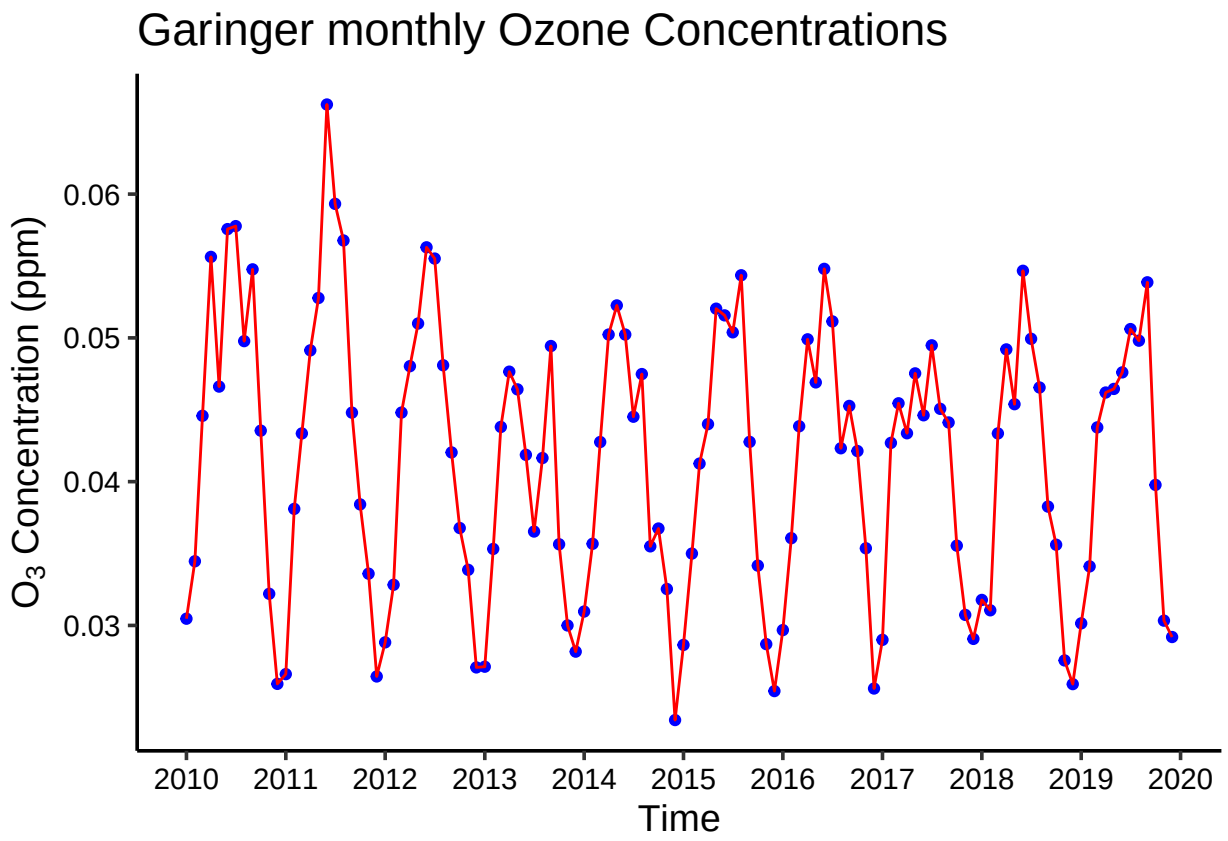
```
# Checking for a monotonic trend using the Seasonal Mann-Kendall test
SMK_test <- SeasonalMannKendall(GaringerOzone.monthly.ts)
print(SMK_test)
```

```
## tau = -0.1, 2-sided pvalue =0.16323
```

Answer: In checking a motonic trend, when a series has a sesaonal pattern, the right test to check the trend is seasonal Mann-Kendall test, and thus, becuase this series shows a clear seasonal pattern throughout the observation period, the seasonal Mann Kendall test is approproaite.

13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. Edit your axis labels accordingly.

```
# Plotting the mean monthly ozone concentrations over time
ggplot(GaringerOzone.monthly, aes(x=Date, y=meanOzone)) +
  geom_point(color="blue")+
  geom_line(color="red", method="lm") +
  scale_x_date(date_breaks = "1 year", date_labels= "%Y") +
  xlab("Time") +
  ylab(expression("O"[3]*" Concentration (ppm)")) +
    ggtitle("Garinger monthly Ozone Concentrations")
```



14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

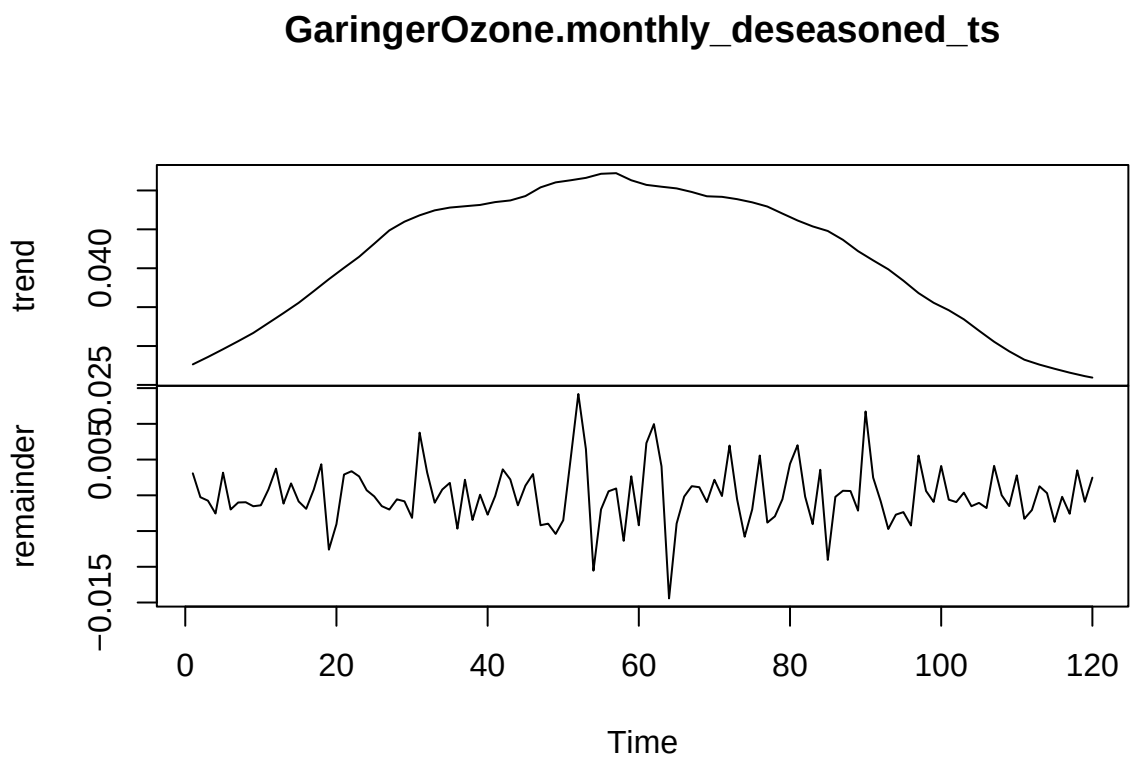
    Answer: The monthly ozone concentrations (in ppm) over the period of 2010 have showed an alternative increasing and decreasing trends over the 12 months Period. The ozone concentrations in ppm in the year 2010 first showed an increasing trend for the first four months (0.03046774, 0.03446429, 0.04458065, 0.05563333 in January, February, March and April respectively). Then the concentration decreased to 0.04661290 in May, and again increased to 0.05756667 in June and a little bit more increament in July (0.05777419). Then the concentration decreased back to 0.04977419 in August and increased once again in September to 0.05476667. In the remaining three months of the year, the concentration continually decreased to 0.04354839 (in October) to 0.03220000 (in Novemebr) and then to 0.02593548 (in December).

15. Subtract the seasonal component from the `GaringerOzone.monthly.ts`. Hint: Look at how we extracted the series components for the EnoDischarge on the lesson Rmd file.

```
# Removing the seasonal component from the monthly time series
GaringerOzone.monthly_deseasoned <- as.data.frame(GaringerOzone.monthly.ts_dec$time.series[,2:3])
```

16. Run the Mann Kendall test on the non-seasonal Ozone monthly series. Compare the results with the ones obtained with the Seasonal Mann Kendall on the complete series.

```
# Conducting the Mann Kendall test
GaringerOzone.monthly_deseasoned_ts <- ts(GaringerOzone.monthly_deseasoned)
plot(GaringerOzone.monthly_deseasoned_ts)
```




```
MK_test <- MannKendall(GaringerOzone.monthly_deseasoned_ts)
print(MK_test)
```

```
## tau = -0.539, 2-sided pvalue =< 2.22e-16
```

Answer: In the Seasonal Mann Kendall test, the p-value is greater than the significance level and thus, we fail to reject the null hypothesis that states the series is stationary. Whereas in the Mann Kendall test on the non-seasonal series, the p-value is less than the significance level alpha and thus, we reject the null hypothesis that states the series is stationary, and therefore, the series follow a trend which is a decreasing trend (tau = -0.539) over time. As Seasonal Mann Kendall test gives the overall trend in a series by looking at and summing up the trend in each month, the resultant trend for the overall series seems to add up to zero referring the series as stationary or has no trend.